

***Amendments to the Claims***

The listing of claims will replace all prior versions, and listings of claims in the application.

1. (currently amended) A method for managing microcode, comprising the steps of:
  - evaluating a mode command to initiate or change a mode;
  - selecting a combination of functions and a sequence list forming a logical sequential concatenation of said functions, each function including microinstructions that, when executed, implement a phase or a sub-phase of said ~~mode~~; mode,
    - wherein said selecting said sequence list includes validating or optimizing said sequence list, one or more of the functions to produce said combination,
    - wherein said validating or optimizing comprises searching for a faster version of a first partial sequence list selectable for said sequence list, wherein said faster version is a second partial sequence list ~~function selected for said combination~~; and
    - delivering said combination to a microcode processor according to said sequence list.
2. (previously presented) A method according to claim 1, wherein said selecting step further comprises the step of:
  - querying a storage medium to select said combination.

3. (previously presented) A method according to claim 1, further comprising the step of:

loading said combination into a microcode instruction memory.

4. (currently amended) A method according to claim 1, further comprising the step of:

loading a said sequence list into a microcode data memory, wherein said sequence list includes a memory address to said combination.

5. (previously presented) A method according to claim 1, further comprising the step of:

executing said combination to implement said mode.

6. (previously presented) A method according to claim 5, further comprising the steps step of:

sending a result from said executing step to a processor for pixel processing or additional microcode processing.

7. (previously presented) A method according to claim 1, further comprising the step of:

sending drawing data to a microcode processor prior to said executing step.

8. (previously presented) A method according to claim 1, further comprising the step of:

sending drawing data to a microcode processor to render three dimensional graphics.

9. (previously presented) A method according to claim 1, further comprising the step of:

sending drawing data to a microcode processor to render an animation scene.

10. (previously presented) A method according to claim 1, further comprising the step of:

sending drawing data to a microcode processor to render a scene for a video game.

11. (currently amended) A system for managing microcode, comprising:  
a mode detector for evaluating a mode command to initiate or change a mode; and

a sequence identifier for selecting a combination of functions and a sequence list forming a logical sequential concatenation of said functions, each function including microinstructions that, when executed, implement a phase or a sub-phase of said mode, wherein said sequence identifier is adapted to validate or optimize said sequence list ~~one or more of the functions to produce said combination~~ by searching for a

faster version of a ~~function selected for said combination~~ first partial sequence list  
selectable for said sequence list, wherein said faster version is a second partial sequence  
list.

12. (previously presented) A system of claim 11, further comprising:  
a code loader for loading said combination into a microcode instruction  
memory.

13. (previously presented) A system of claim 11, further comprising:  
a phase executor for commanding a microcode processor to execute said  
combination.

14. (previously presented) A system of claim 11, further comprising:  
a drawing data processor for sending drawing data or input for drawing  
data to a microcode processor in response to said mode command.

15. (previously presented) A system of claim 11, further comprising:  
a drawing data processor for sending drawing data or input for drawing  
data to a microcode processor to render a three dimensional model in response to said  
mode command.

16. (previously presented) A system of claim 11, further comprising:  
a drawing data processor for sending drawing data or input for drawing data to a microcode processor to render an animation scene in response to said mode command.
17. (currently amended) A system of claim 11, further comprising:  
a microcode data memory for storing ~~a~~said sequence list specifying a memory address to said combination.
18. (currently amended) A computer program product comprising a computer useable medium having computer readable program code means embedded in said medium for causing an application program to execute on a computer used to manage microcode, said computer readable program code means comprising:  
first computer readable program code means for causing the computer to evaluate a mode command to initiate or change a mode;  
second computer readable program code means for causing the computer to select a combination of functions and a sequence list forming a logical sequential concatenation of said functions, each function including microinstructions that, when executed, implement a phase or a sub-phase of said mode; and  
third computer readable program code means for causing the computer to produce a validated or optimized said sequence list ~~validate or optimize one or more of the functions to produce said combination~~, wherein said third computer readable program code means includes computer readable program code means for causing the

computer to search for a faster version of a first partial sequence list selectable for said sequence list, wherein said faster version is a second partial sequence list-function  
~~selected for said combination.~~

19. (previously presented) A computer program product according to claim 18, wherein said second computer readable program code means loads said combination into a microcode instruction memory.

20. (previously presented) A computer program product according to claim 18, further comprising:

fourth computer readable program code means for causing the computer to command a microcode processor to execute said combination.

21. (previously presented) A computer program product according to claim 18, further comprising:

fourth computer readable program code means for causing the computer to send drawing data or input for drawing data to a microcode processor in response to said mode command.

22. (previously presented) A computer program product according to claim 18, further comprising:

fourth computer readable program code means for causing the computer to send drawing data or input for drawing data to a microcode processor to render three-dimensional graphics in response to said mode command.

23. (currently amended) A computer program product according to claim 18, further comprising:

fourth computer readable program code means for causing the computer to store a said sequence list specifying a memory address to said combination.

24. (currently amended) A method for managing microcode, comprising the steps of:

accessing a library of functions, each function including microinstructions that, when executed, implement a phase or a sub-phase of a graphics mode;

selecting a combination of functions from said library and a sequence list forming a logical sequential concatenation of said functions in response to a mode command to produce a desired mode ~~mode~~; mode,

wherein said selecting said sequence list includes validating or optimizing said sequence list, ~~one or more of the functions to produce said combination~~, wherein said validating or optimizing comprises searching for a faster version of a first partial sequence list selectable for said sequence list, wherein said faster version is a second partial sequence list ~~function selected for said combination~~;

delivering said combination to a processor;  
delivering drawing data to said processor; and  
executing said combination to process said drawing data according to said sequence list and thereby render said desired mode.

25. (previously presented) A method according to claim 24, wherein said validating or optimizing step further comprises:

selecting a merger group from said library, wherein said merger group includes a combination of microinstructions that, when executed, implement a plurality of phases of a graphics mode.

26. (previously presented) A method according to claim 24, wherein said validating or optimizing step further comprises:

preprocessing data for said combination to calculate values used repetitively during said executing step.

27. (previously presented) A method according to claim 24, wherein said validating or optimizing step further comprises:

validating a loading state of a function selected for said combination.

28. (cancelled)



29. (previously presented) A method according to claim 24, wherein said validating or optimizing step further comprises:
- validating one or more of the functions to produce said combination.